

In the Claims

Please amend claims 16, 22 and 25. A complete listing of all claims and status indicators is set forth below. This listing of claims will replace all prior versions and listings of claims in the application.

1. (previously presented) A method for storing transaction entries in a transaction order queue, comprising the acts of:
 - a) temporarily storing a plurality of transaction entries;
 - b) prioritizing each of the plurality of temporarily stored transaction entries according to a bus standard;
 - c) selecting one of the plurality of temporarily stored transaction entries; and
 - d) enqueueing the selected one of the plurality of temporarily stored transaction entries into one of a plurality of storage locations in the transaction order queue according to an associated priority.
2. (original) The method as recited in claim 1, wherein act (a) comprises the act of temporarily storing a plurality of transaction entries in a bank of registers.
3. (original) The method as recited in claim 1, wherein the plurality of transaction entries are temporarily stored simultaneously.
4. (previously presented) The method as recited in claim 1, wherein acts (c) and (d) comprise the acts of:

determining whether a posted write transaction entry is present;

if the posted write transaction entry is present, then enqueueing the posted write transaction entry into the transaction order queue,

if the posted write transaction entry is not present, then determining whether a read completion transaction entry is present;

if the read completion transaction entry is present, then enqueueing the read completion transaction entry into the transaction order queue;

if the read completion transaction entry is not present, then determining whether a delayed/split transaction entry is present; and

if the delayed/split transaction entry is present, then enqueueing delayed/split transaction entry into the transaction order queue.

5. (original) The method as in claim 1, comprising the act of enqueueing each of the plurality of transaction entries into the transaction order queue one at a time during successive clock cycles.

6. (previously presented) A method of manufacturing a computer system for storing transaction entries in a transaction order queue comprising the acts of:

providing a temporary storage space for temporarily storing a plurality of transaction entries;

providing logic to prioritize each of the plurality of transaction entries according to a bus standard;

providing logic to select one of the plurality of temporarily stored transaction entries; and

providing a transaction order queue comprising a plurality of storage locations, wherein

each of the plurality of storage locations is configured to receive the selected one of

the plurality of temporarily stored transaction entries according to an associated priority determined by the prioritization logic.

7. (original) The method as recited in claim 6, wherein the act of providing temporary storage space for temporarily storing a plurality of transaction entries comprises the act of providing a plurality of registers.

8. (original) The method as recited in claim 7, wherein the act of providing logic to select one of the plurality of temporarily stored transaction entries comprises the act of providing logic for:

determining whether a posted write transaction entry is present;

if the posted write transaction entry is present, then enqueueing the posted write transaction entry into the transaction order queue;

if the posted write transaction entry is not present, then determining whether the transaction is a read completion transaction entry;

if the read completion transaction entry is present, then enqueueing the read completion transaction entry into the transaction order queue;

if the read completion transaction entry is not present, then determining whether a delayed/split transaction entry is present; and

if the delayed/split transaction entry is present, then enqueueing the delayed/split transaction entry into the transaction order queue.

9. (previously presented) An system for providing multiple simultaneous transaction entries to a transaction order queue comprising:

means for temporarily storing simultaneously a plurality of transaction entries

simultaneously;

means for prioritizing each of the plurality of transaction entries according to a bus

standard;

means for selecting one of the plurality of temporarily stored transaction entries; and

means for enqueueing the selected one of the plurality of temporarily stored transaction

entries in one of a plurality of storage locations of a transaction order queue

according to an associated priority determined by the prioritization means.

10. (original) The system as in claim 9, wherein the means for temporarily storing a plurality of transaction entries comprises a bank of registers.

11. (original) The system as in claim 9, wherein the means for selecting one of the plurality of temporarily stored transaction entries comprises:

means for determining whether a posted write transaction entry is present;

means for enqueueing the posted write transaction entry into the transaction order queue if the posted write transaction entry is present;

means for determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;

means for enqueueing the read completion transaction entry into the transaction order queue if the read completion transaction entry is present;

means for determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

means for enqueueing the delayed/split transaction entry into the transaction order queue if

the delayed/split transaction entry is present.

12. (previously presented) A system for providing multiple simultaneous transaction entries to a transaction order queue comprising:

a temporary memory storage adapted to store a plurality of transaction entries; and
logic adapted for selecting and ordering the plurality of transaction entries in the transaction order queue according to a bus standard from the temporary memory storage for processing, wherein the transaction order queue comprises a plurality of storage locations.

13. (original) The system as in claim 12, wherein the temporary memory storage comprises a bank of registers.

14. (original) The system as in claim 12, wherein the logic comprises:
logic adapted for determining whether a posted write transaction entry is present;
logic adapted for enqueueing the posted write transaction entry into the transaction order queue if the posted write transaction entry is present;
logic adapted for determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;
logic adapted for enqueueing the read completion transaction entry into the transaction order queue if the read completion transaction entry is present;
logic adapted for determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

logic adapted for enqueueing the delayed/split transaction entry into transaction order queue
if the delayed/split transaction entry is present.

15. (original) The method as in claim 12, wherein each of the plurality of transaction entries is enqueueued into the transaction order queue one at a time during successive clock cycles.

16. (currently amended) A processing system comprising:
a first logic device;
a plurality of registers being configured to receive a plurality of transaction entries from the first logic device;
a second logic device adapted to receive the transaction entries from the plurality of registers and being programmed to ~~prioritize~~select transaction entries according to PCI-X specifications; ~~and~~
a third logic device adapted to select the transaction entries from the plurality of registers according to an associated priority; and
a transaction order queue comprising a plurality of storage locations, wherein the third logic device is configured to enqueue the transaction entries into the transaction order queue according to the associated priority, each of the plurality of storage locations being configured to receive and enqueue the selected transaction entries.

17. (original) The processing system as in claim 16, wherein the first logic device receives transaction entries from an input source.

18. (original) The processing system as in claim 16, wherein the plurality of registers store the plurality of transaction entries received from the first logic device.
19. (original) The processing system as in claim 16, wherein the second logic device selects a single entry to send to the transaction order queue.
20. (original) The processing system as in claim 19, wherein the second logic device comprises:
- logic adapted for determining whether a posted write transaction entry is present;
 - logic adapted for enqueueing the posted write transaction entry into the transaction order queue if the posted write transaction entry is present;
 - logic adapted for determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;
 - logic adapted for enqueueing the read completion transaction entry into the transaction order queue if the read completion transaction entry is present;
 - logic adapted for determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and
 - logic adapted for enqueueing the delayed/split transaction entry into the transaction order queue if the delayed/split transaction entry is present.
21. (original) The processing system as in claim 16, wherein the transaction order queue enqueues one transaction entry per clock cycle.
22. (currently amended) A computer system comprising:
- at least one processor;

a memory device operatively coupled to the at least one processor; and
a transaction order queue circuit configured to process transactions from the memory device, the transaction order queue circuit being adapted to encode a plurality of simultaneously received transaction entries according to an associated priority, and comprising a transaction order queue comprising a plurality of storage locations, wherein each of the plurality of storage locations is configured to store one of the transactions from the memory device, in an order according to the associated priority.

23. (original) The computer system as in claim 22, wherein the transaction order queue comprises:
logic adapted for determining whether a posted write transaction entry is present;
logic adapted for enqueueing the posted write transaction entry into the transaction order queue, if the posted write transaction entry is present;
logic adapted for determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;
logic adapted for enqueueing the read completion transaction entry into the transaction order queue, if the read completion transaction entry is present;
logic adapted for determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and
logic adapted for enqueueing the delayed/split transaction entry into the transaction order queue if the delayed/split transaction entry is present.

24. (original) The system as in claim 22, wherein the computer system comprises network capabilities.

25. (currently amended) A method for storing transaction entries in a transaction order queue, comprising the acts of:

- a) temporarily storing a plurality of simultaneously received transaction entries;
- and
- b) delivering the plurality of transaction entries to a transaction order queue one at a time, wherein each of the plurality of transaction entries is stored in a respective one of a plurality of storage locations in the transaction order queue, in an order according to an associated priority.

26. (original) The method as in claim 25, wherein the plurality of simultaneous transaction entries is stored in a bank of registers.

27. (original) The method as in claim 25, wherein act (b) comprises the act of:

determining whether a posted write transaction entry is present;

enqueueing the posted write transaction entry into the transaction order queue, if the posted write transaction entry is present;

determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;

enqueueing the read completion transaction entry into the transaction order queue, if the read completion transaction entry is present;

determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and

enqueuing the delayed/split transaction entry into the transaction order queue.

28. (previously presented) A method for storing transaction entries in a transaction order queue, comprising the acts of:

- a) temporarily storing a plurality of simultaneously received transaction entries;
- b) prioritizing each of the temporarily stored transaction entries; and
- c) transmitting the stored transaction entries to the transaction order queue

according to priority, wherein each of the stored transaction entries is transmitted to a respective one of a plurality of storage locations in the transaction order queue.

29. (original) The method as in claim 28, wherein the plurality of transaction entries are stored simultaneously in a bank of registers.

30. (original) The method as in claim 28, wherein act (b) comprises the acts of:
determining whether a posted write transaction entry is present;
enqueuing the posted write transaction entry into the transaction order queue, if the posted write transaction entry is present;
determining whether a read completion transaction entry is present, if the posted write transaction entry is not present;
enqueuing the read completion transaction entry into the transaction order queue, if the read completion transaction entry is present;
determining whether a delayed/split transaction entry is present, if the read completion transaction entry is not present; and
enqueuing the delayed/split transaction entry into the transaction order queue.